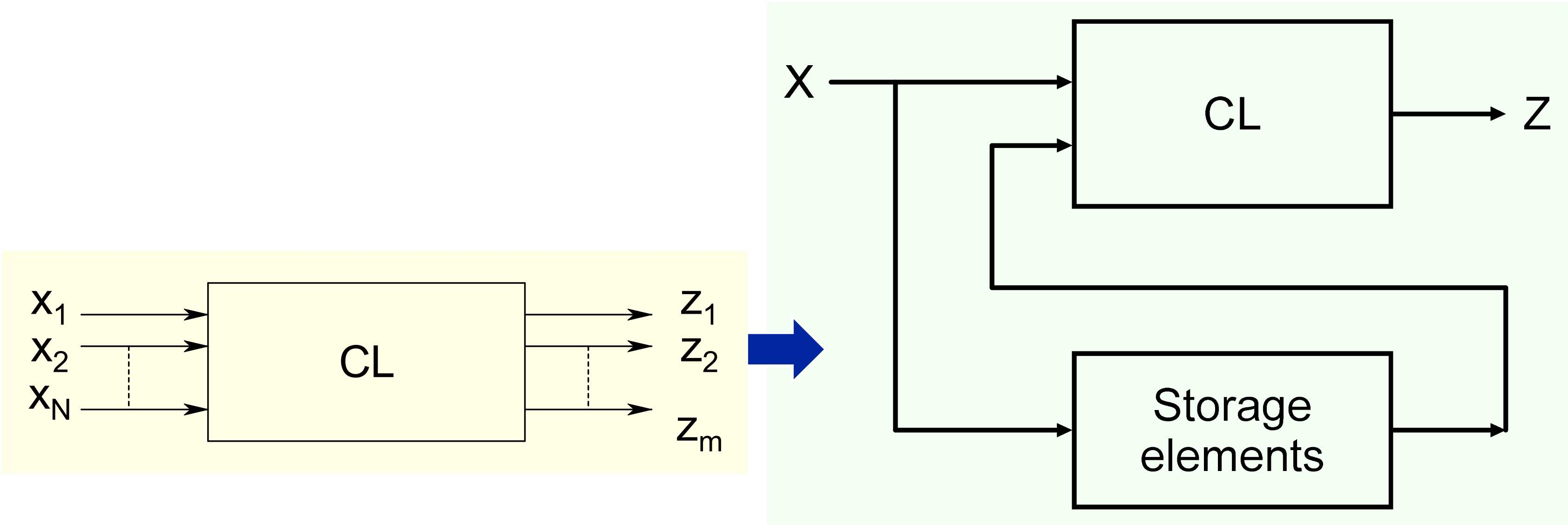
ESC201T : Introduction to Electronics

Lecture 38: Sequential circuit design-2

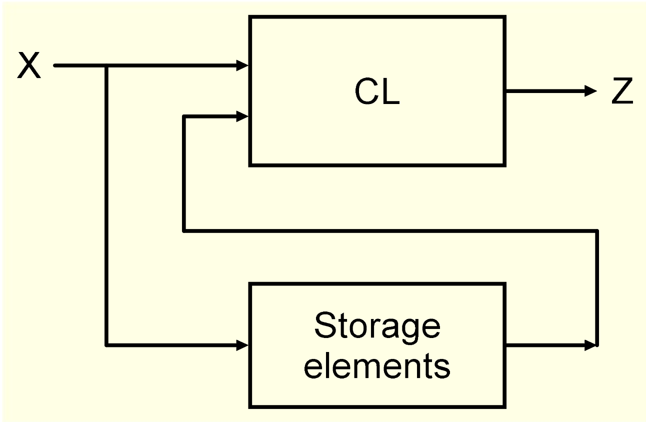
B. Mazhari
Dept. of EE, IIT Kanpur

Limitations of Combinational logic:
decisions can only be based on the present value of inputs

- A more general purpose decision making machine should be able to make decisions based on past values of inputs as well.



Limitations:



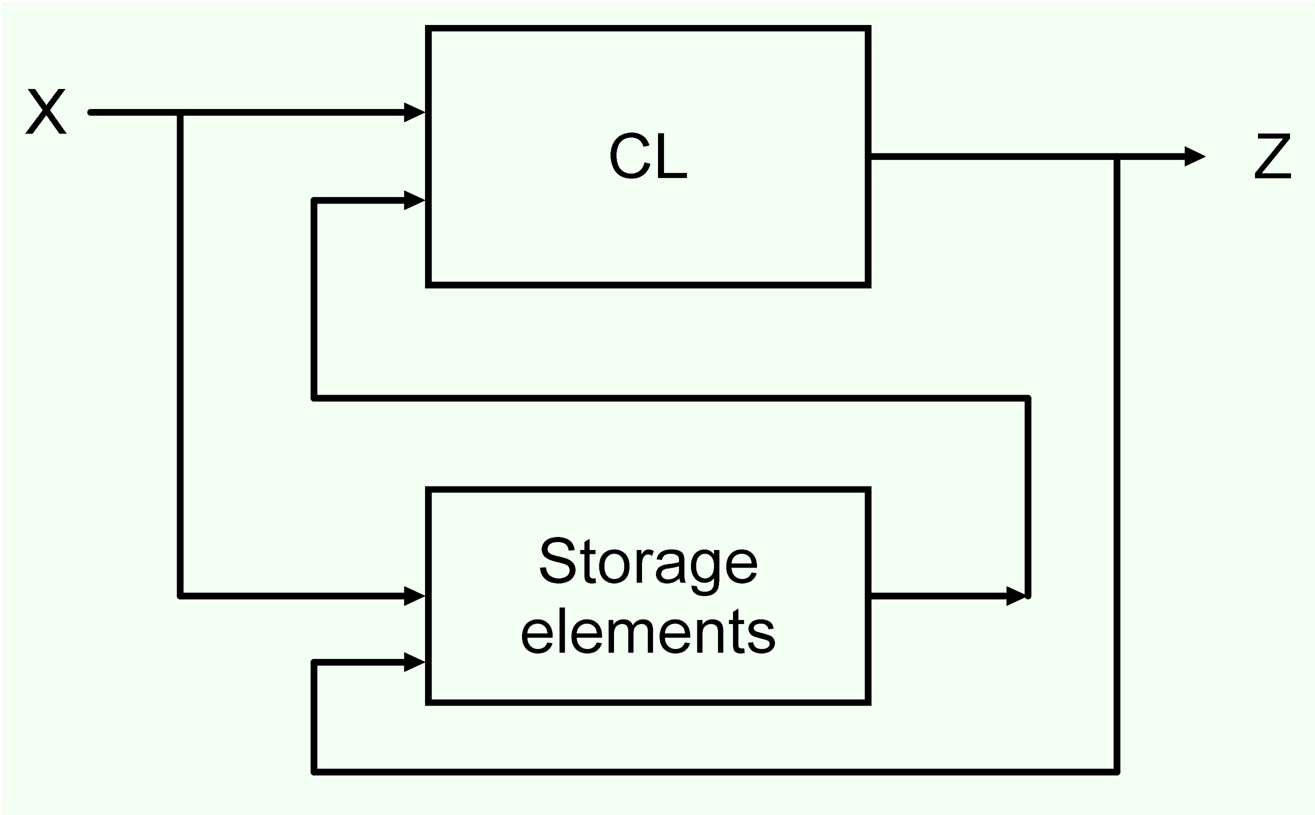
$z[n] = x[n]$ .OR. $z[n-1]$

$z[n] = x[n]$ .OR. $x[n-1]$ .OR. $x[n-2]$……..$x[0]$

Requires infinite memory !

$\Rightarrow$Make provision for storage of past values of Outputs as well

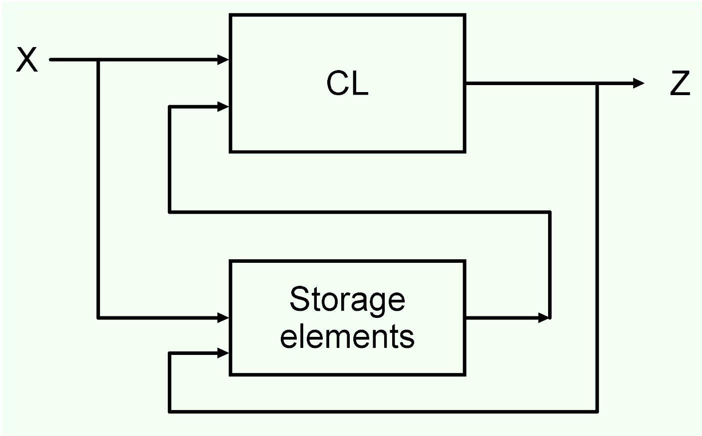
Improved System:



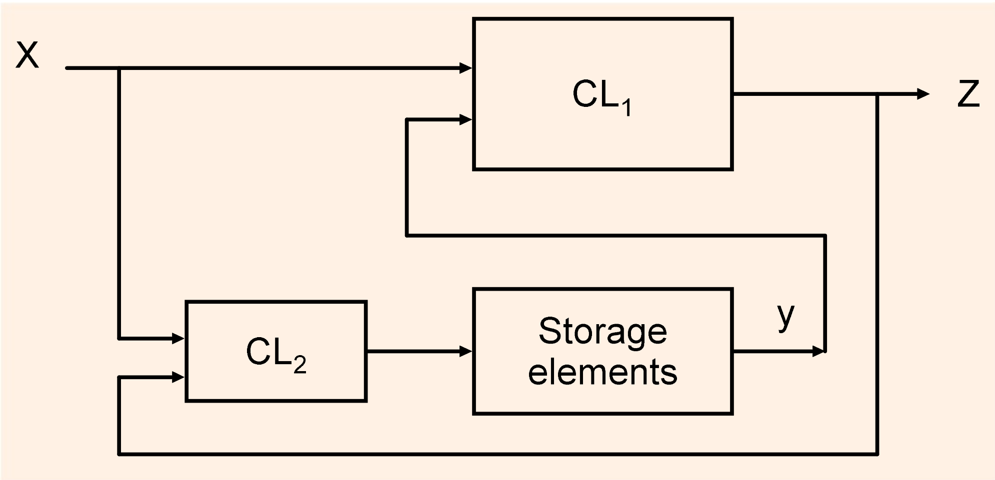
Limitations:

$z[n] = x[n]$ .OR. $(x[n-1]$ .AND. $z[n-1])$

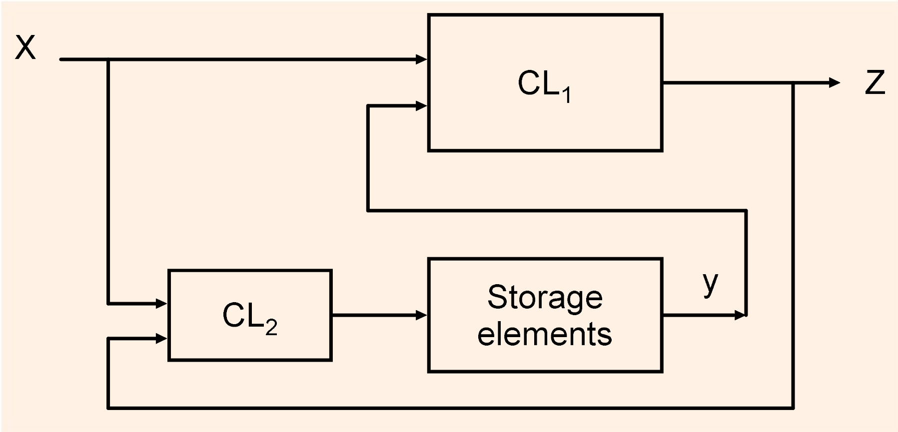
Requires two storage elements



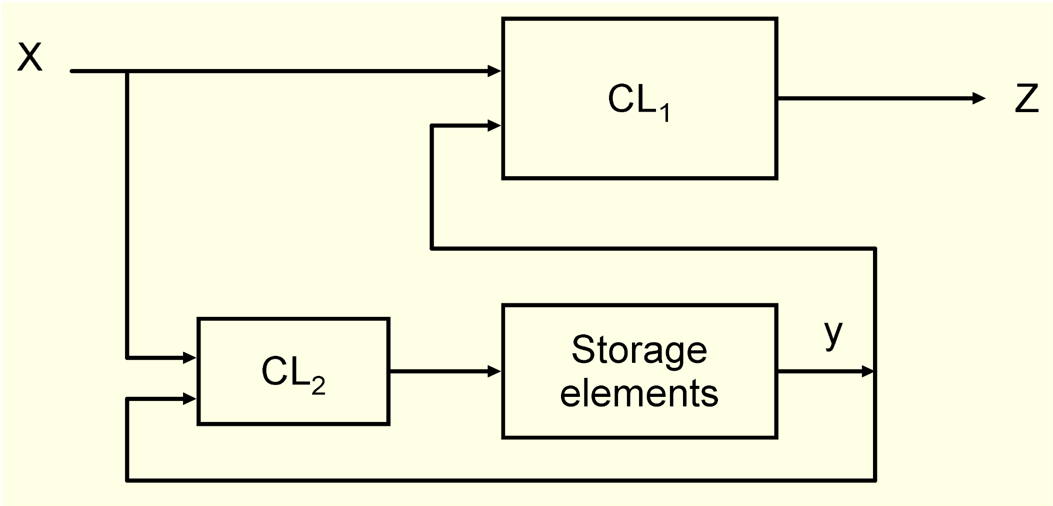
However, by defining a new variable
$y[n] = x[n-1]$.AND. $z[n-1]$, we can use only one storage element.

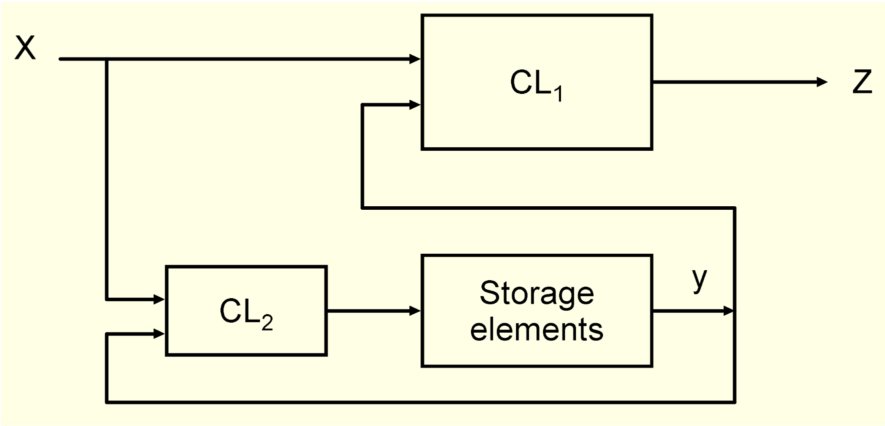


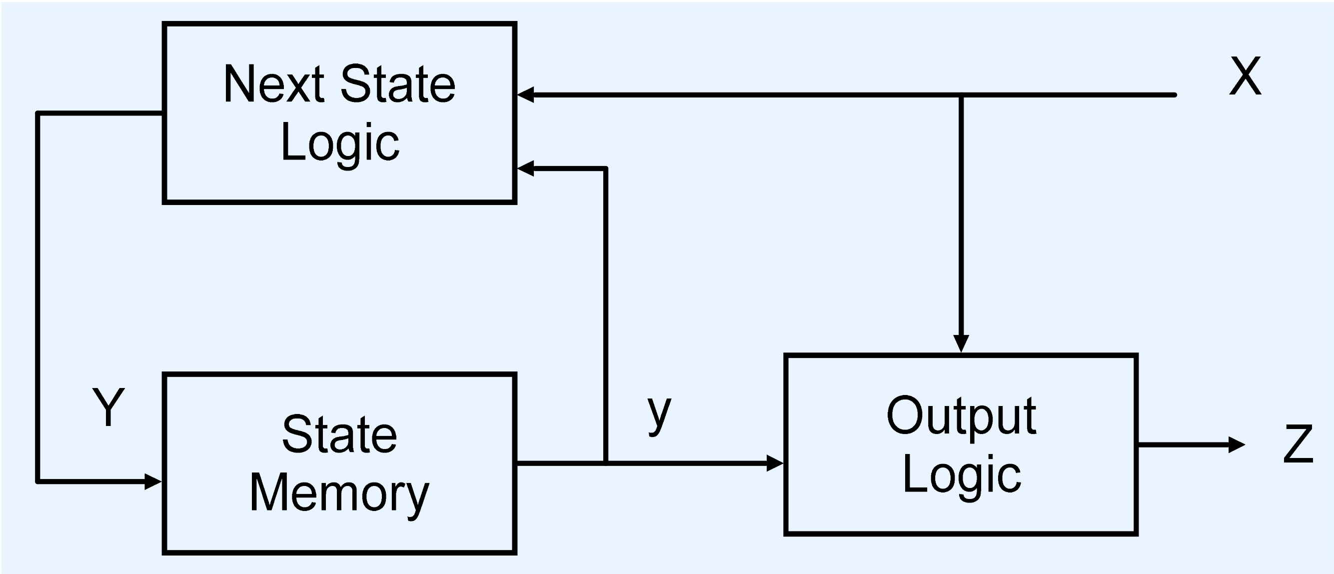
Improved Decision Making Machine:



Since Output Z is a function of Y and X

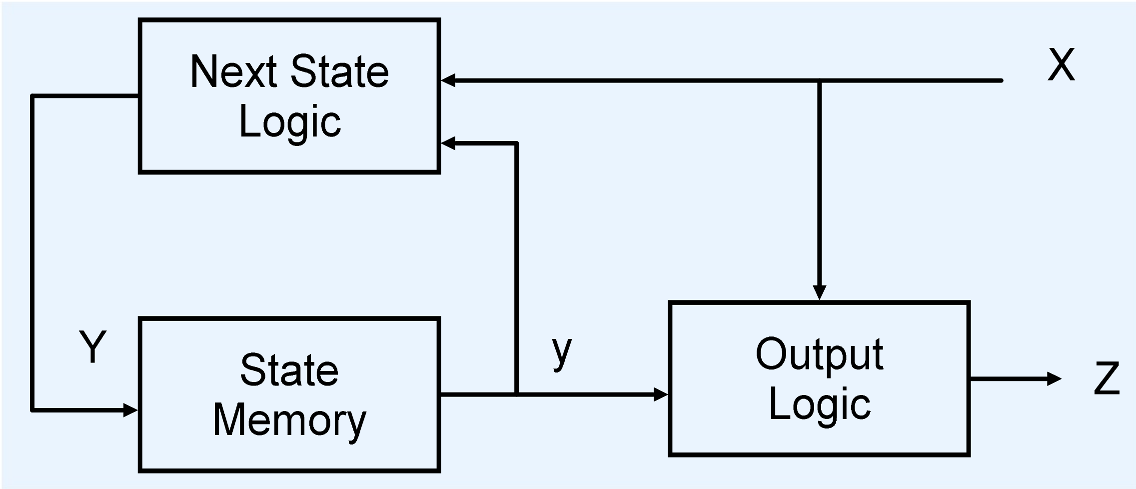


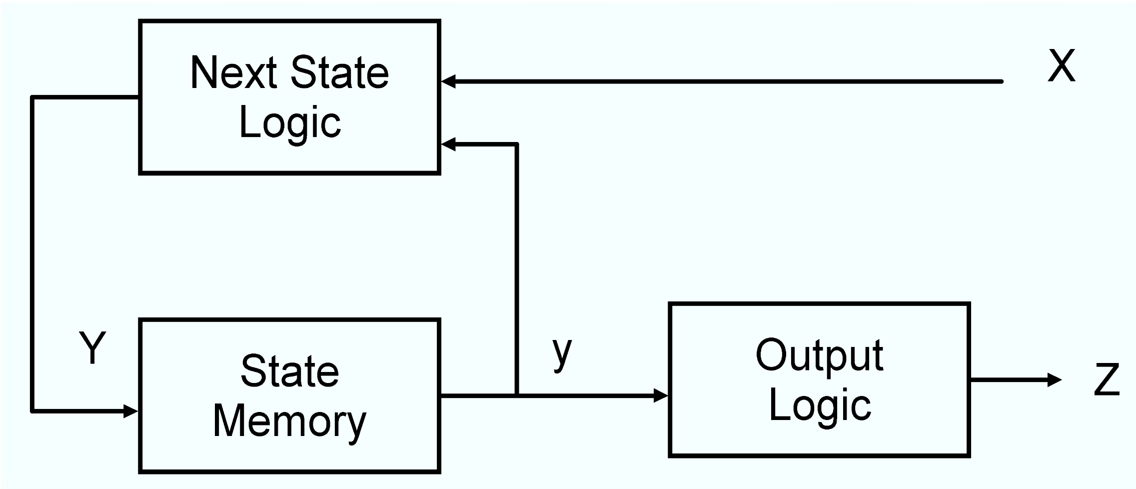




Mealy Sequential Machine

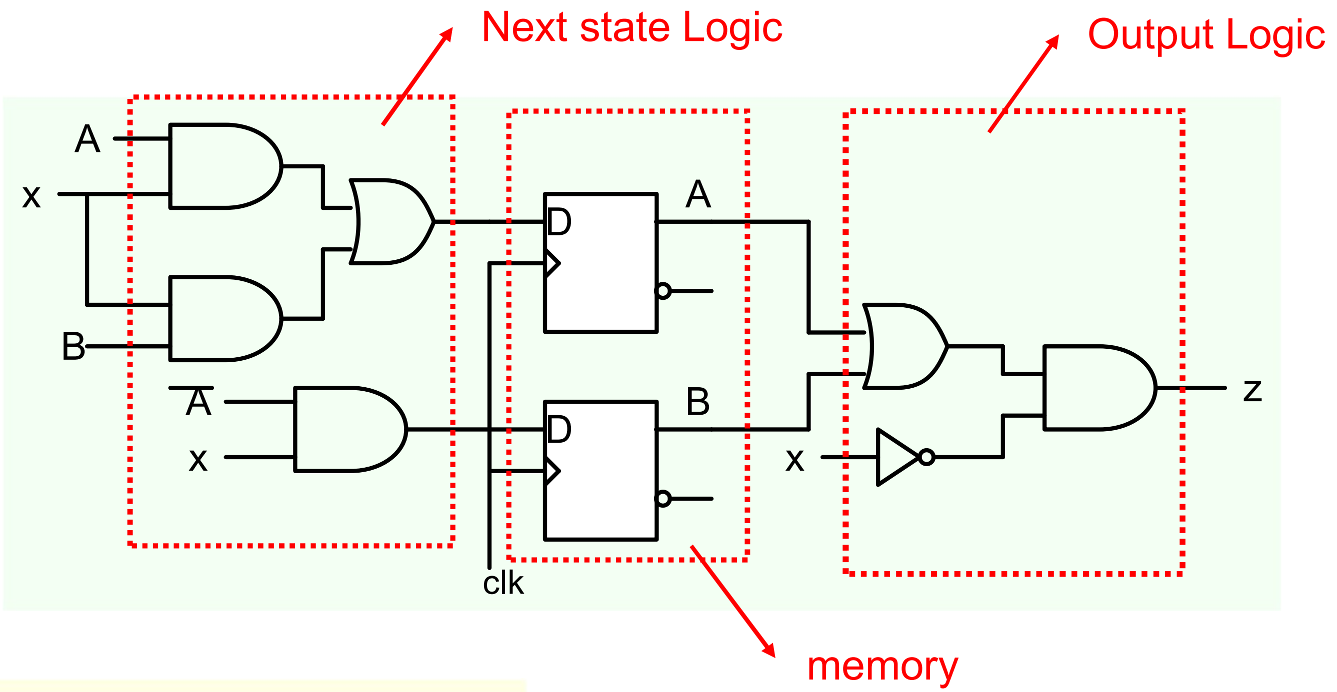
y: Present State
Y : Next State

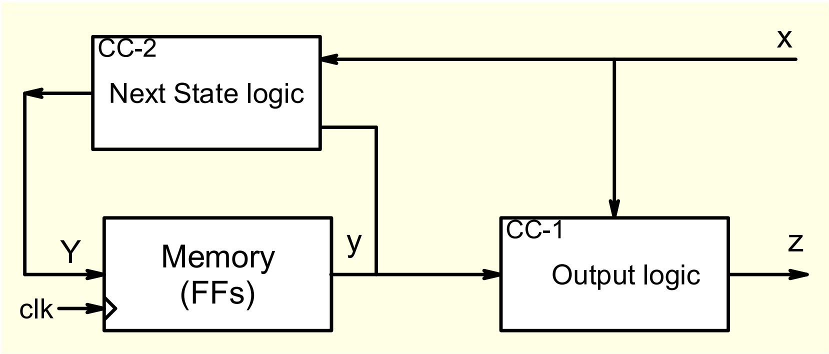




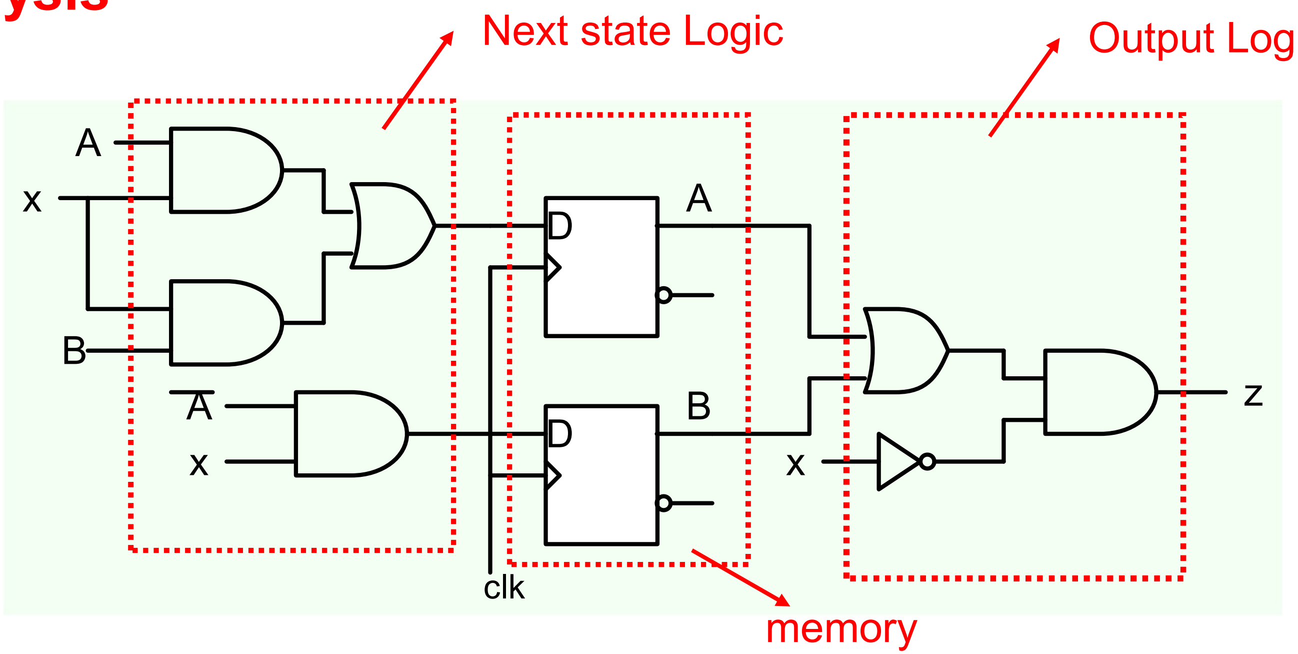
Moore Sequential Machine

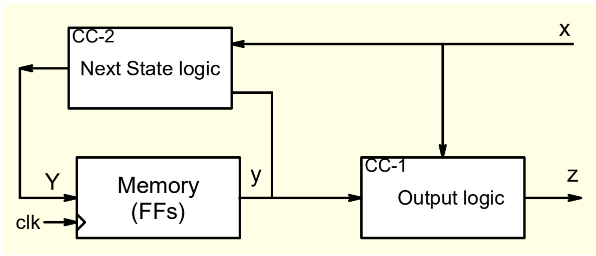
Example of a synchronous sequential circuit





Analysis





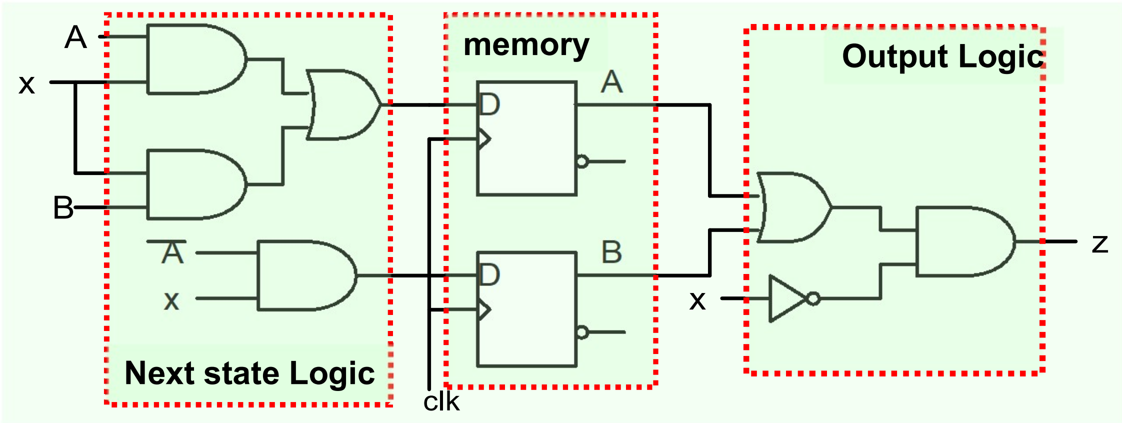
The dependence of output z on input x depends on the state of the memory (A,B)

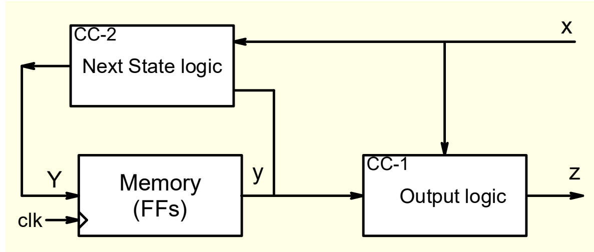
The memory has 2 FFs and each FF can be in state 0 or 1. Thus there are four possible states: AB: 00,01,10,11.

To describe the behavior of a sequential circuit, we need to show

1. how the system goes from one memory state to the next as the input changes
2. How the output responds to input in each state

Analysis of Sequential Circuits





$$D_A = A.x + B.x \quad ; D_B = \overline{A}.x ; z = (A+B).\overline{x}$$

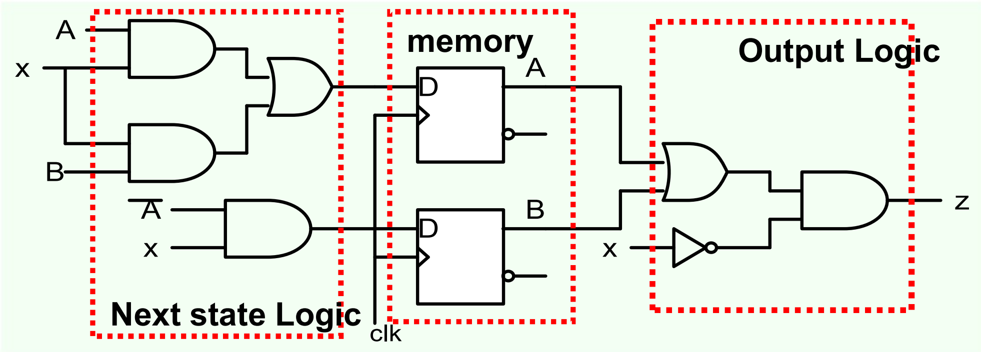
$$A(t+1) = A(t).x + B(t).x$$

$$B(t+1) = \overline{A(t)}.x$$

$$z = (A+B).\overline{x}$$

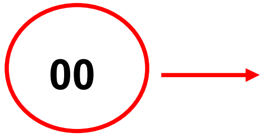
State Transition Table

Present State		Input	Next State		Output
A	B	x	A	B	z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

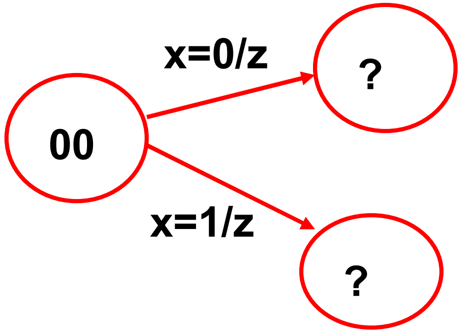


State Transition Table

Present State A	Present State B	Input x	Next State A	Next State B	Output z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

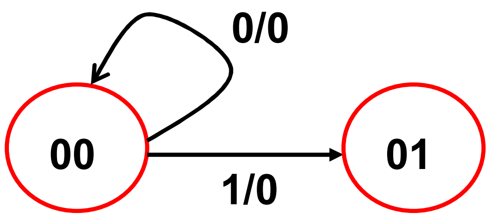


Memory state in which FF A& B have output values 00

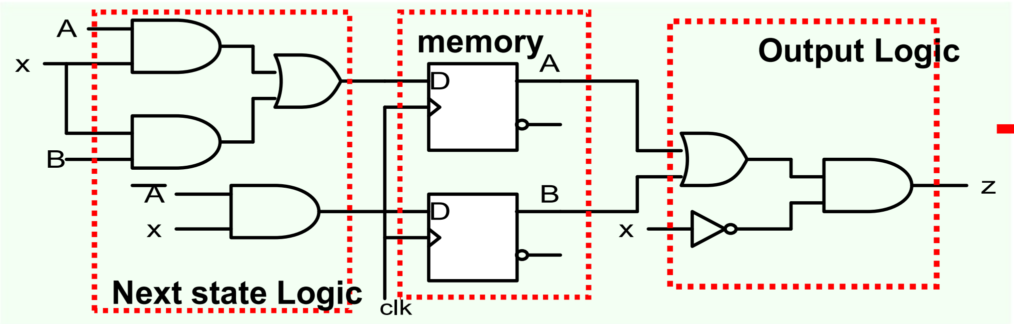


If x = 0 then z = 0, When the clock edge comes the system would stay in 00 state.

If x = 1 then z = 0. When the clock edge comes the system would go to 01 state.



Analysis of Sequential Circuits



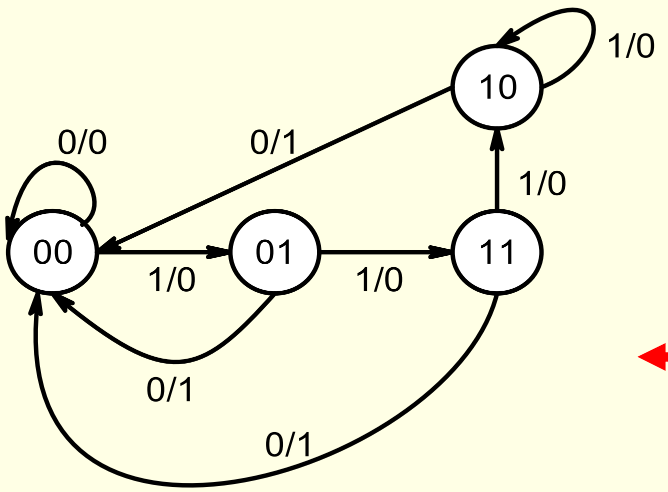
$$A(t+1) = A(t).x + B(t).x$$

$$B(t+1) = \overline{A(t)}.x$$

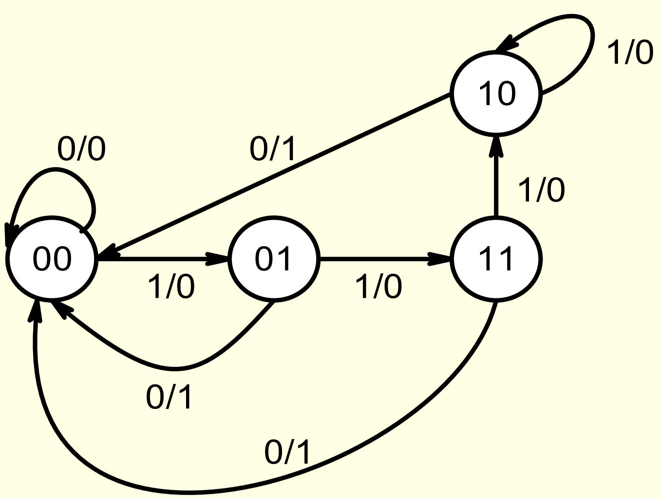
$$z = (A+B).\overline{x}$$

State Transition Table

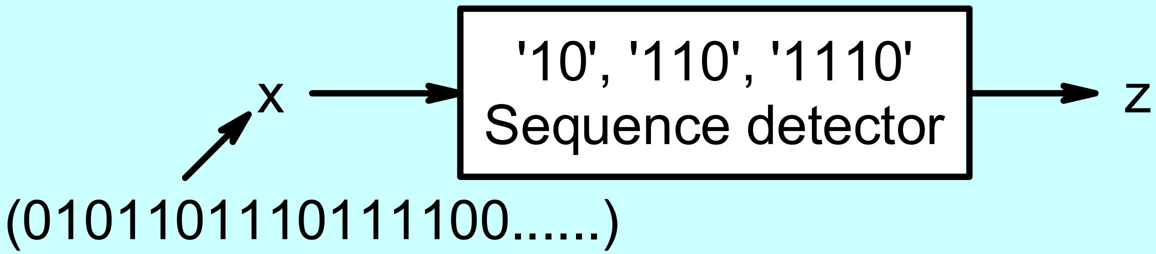
Present State A	Present State B	Input x	Next State A	Next State B	Output z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



State transition Graph



State transition Graph



Design of Sequential Circuits

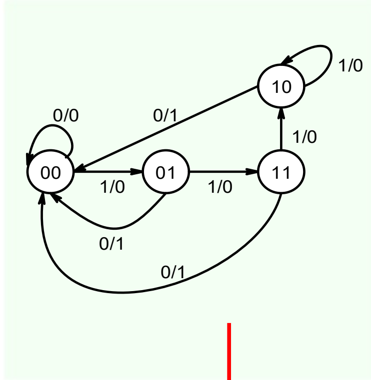
Specifications

State Transition Graph

Choose FF State Encoding

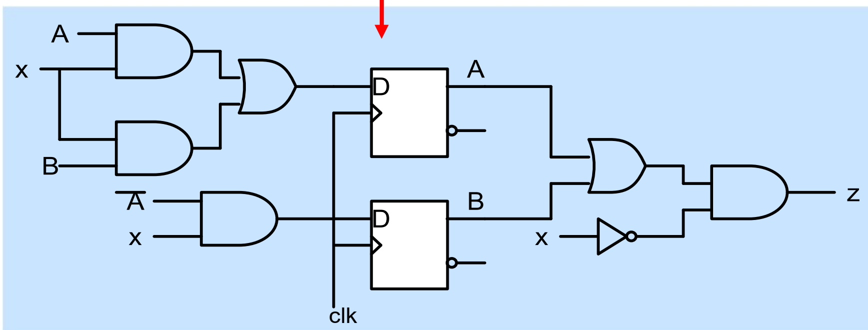
State Table

Synthsize Combinational Circuits

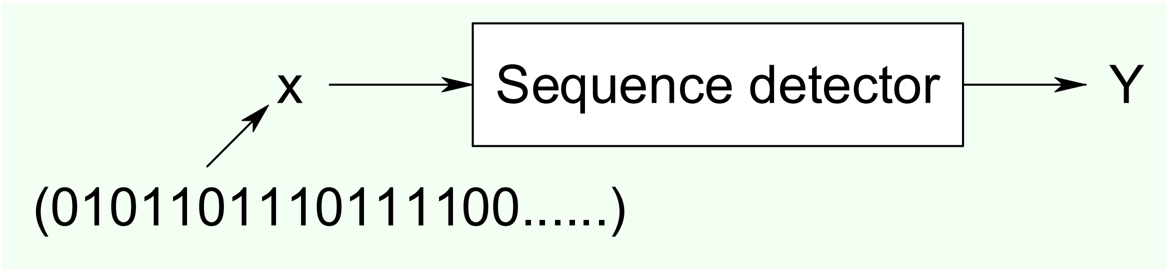


State Transition Table

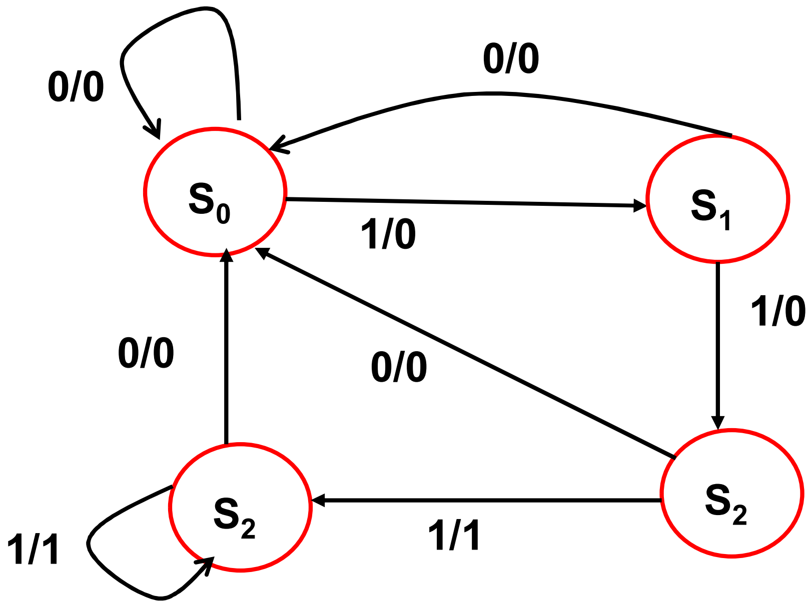
Present State A	Present State B	Input x	Next State A	Next State B	Output z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0



System specification to State Transition Graph

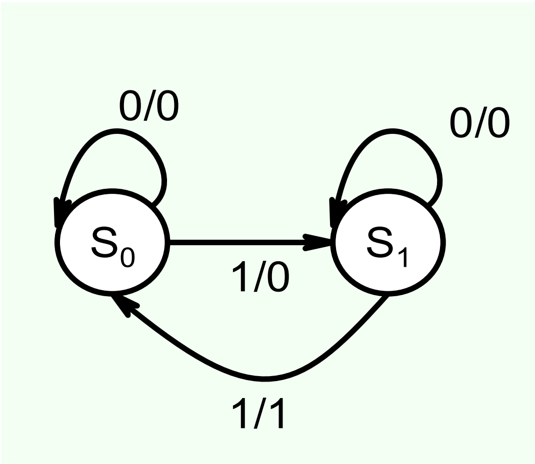


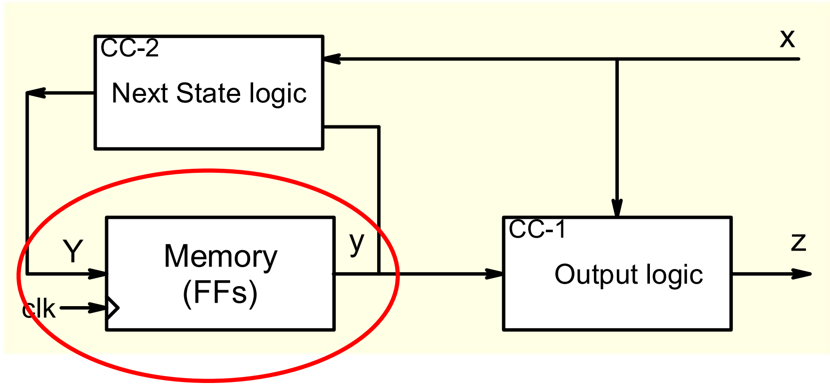
Detect 3 or more consecutive 1's in the input stream



Conversion of State transition graph to a circuit

Example-1



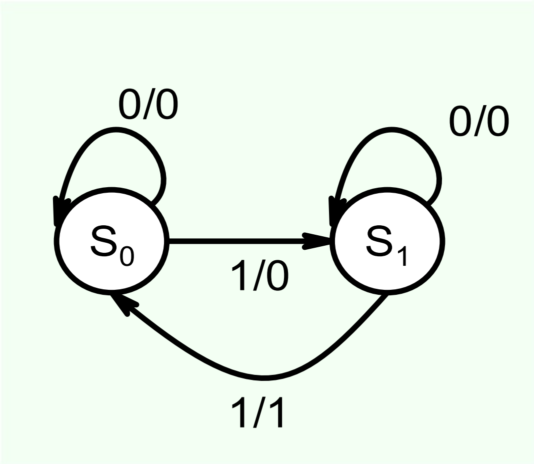


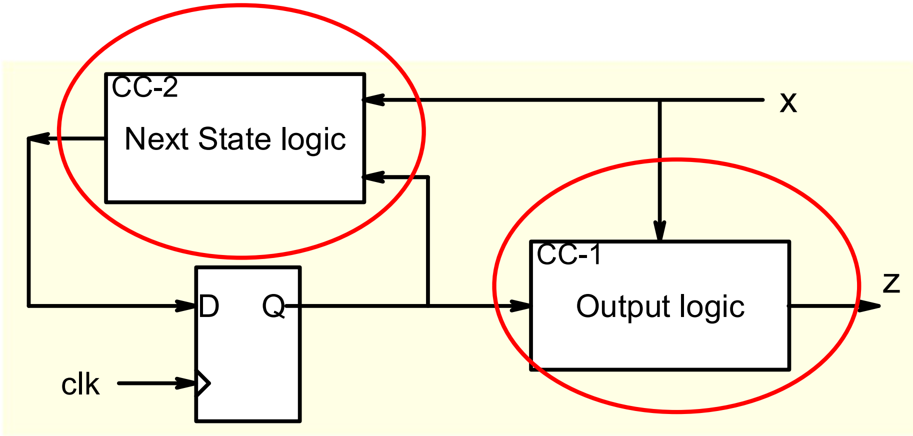
3 blocks need to be designed

1. How many FFs do we need? **N FFS can represent 2^N states so Minimum is 1**

2. Which FF do we choose? **Say D FF**

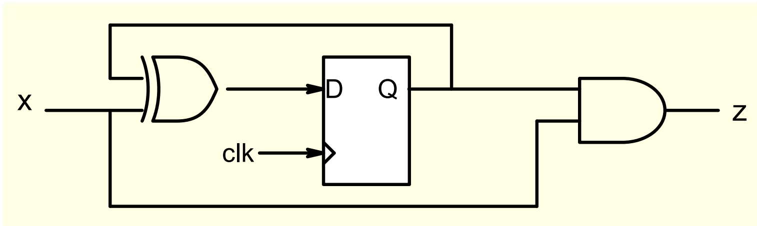
3. How are the states encoded? **Say FF output Q=0 represents S_0 and Q=1 represents S_1 state**





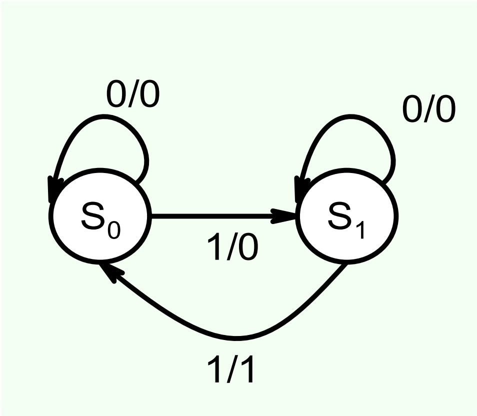
State Transition Table

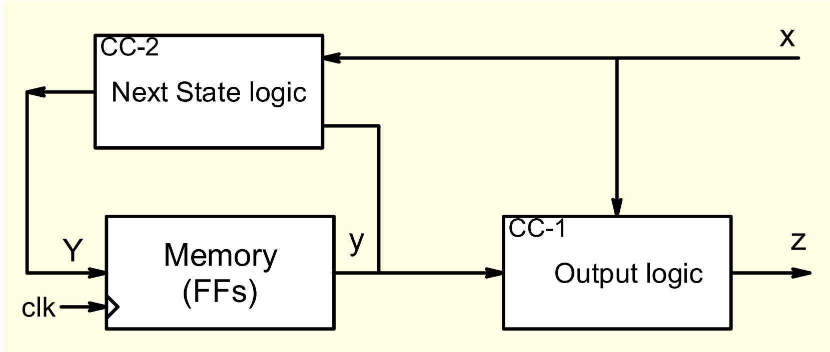
Present State Q(t)	Input x	Next State Q(t+1)	D	Output z
0 0	0 1	0 1	**0** **1**	0 0
1 1	0 1	1 0	**1** **0**	0 1



$$D = \overline{Q}.x + Q.\overline{x} \quad ; \quad z = Q.x$$

Example-2

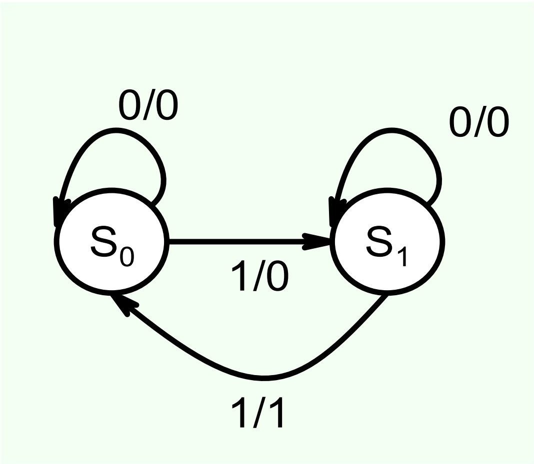


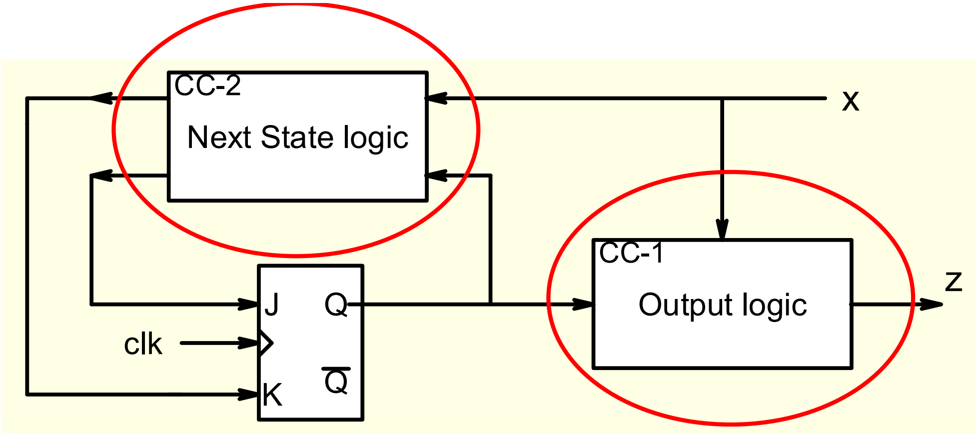


1. How many FFs do we need? **1**

2. Which FF do we choose? **Say JK FF**

3. How are the states encoded? **Say FF output Q=0 represents S_0 and Q=1 represents S_1 state**

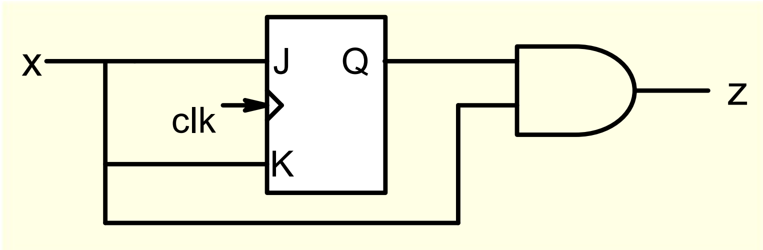




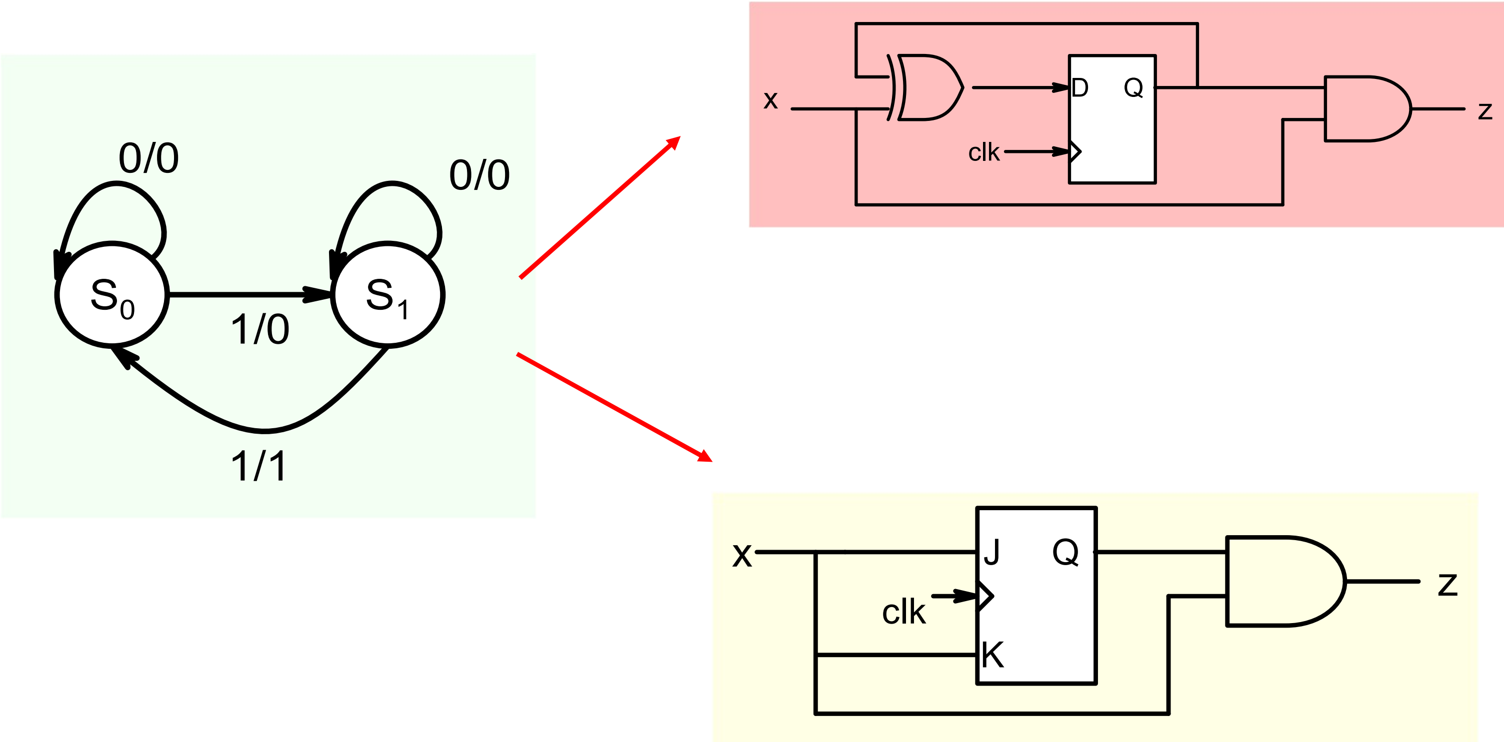
Q(t)	Q(t+1)	J K
0	0	0 X
0	1	1 X
1	0	X 1
1	1	X 0

State Transition Table

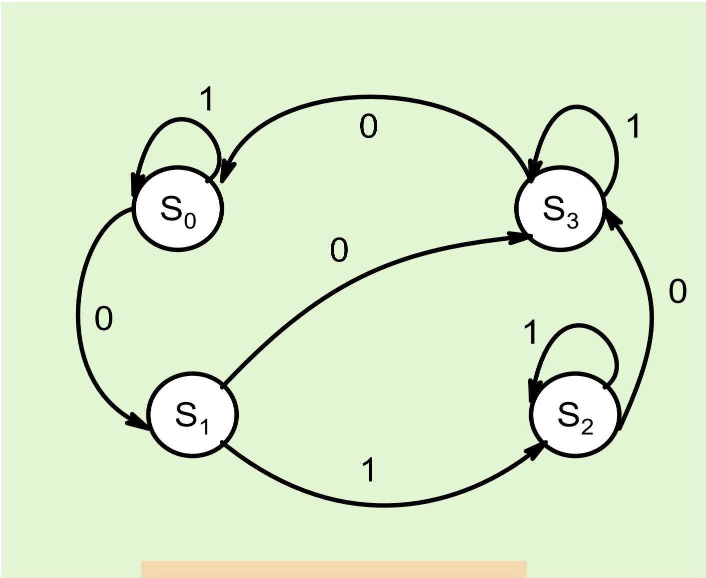
Present State Q(t)	Input x	Next State Q(t+1)	J	K	Output z
0	0	0	**0**	**X**	0
0	1	1	**1**	**X**	0
1	0	1	**X**	**0**	0
1	1	0	**X**	**1**	1



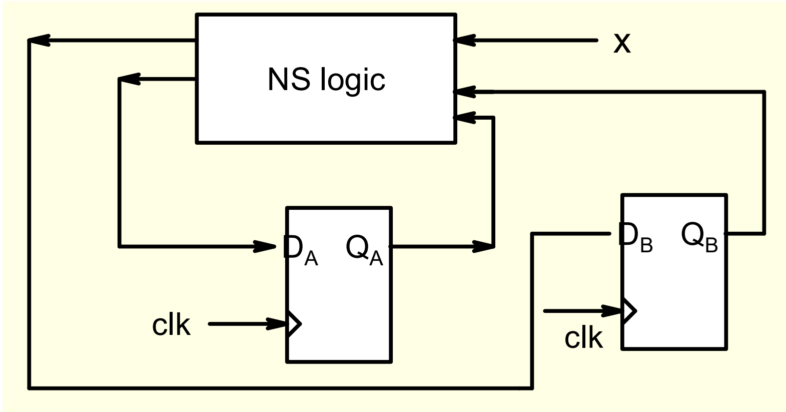
$$J = x \;\; ; K = x; \;\; z = Q.x$$


0/0
0/0
S0
S1
1/0
1/1
x
D
Q
clk
z
x
J
Q
clk
K
z

Example-3



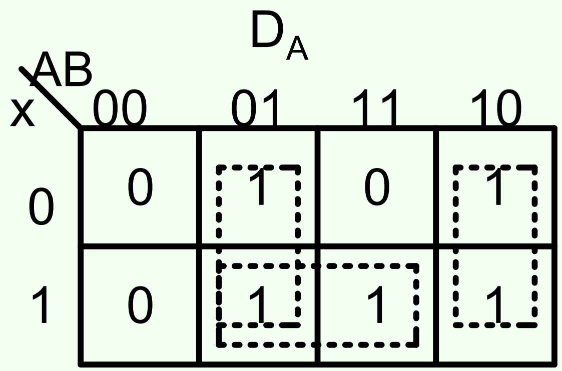
For 4 states a minimum of two FFs will be required. Let us choose 2 D FFs A &B



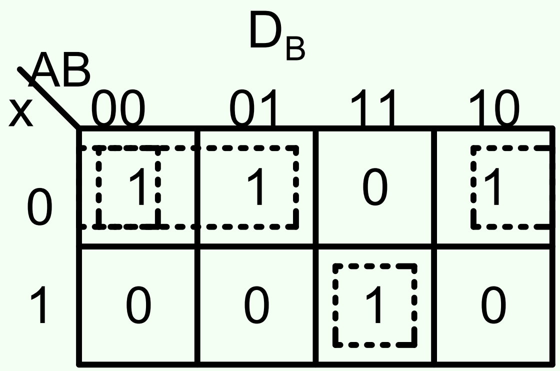
State	FF O/P	
	A	B
S_0	0	0
S_1	0	1
S_2	1	0
S_3	1	1

Present State		Input	Next State			
A	B	x	A	B	D_A	D_B
0	0	0	0	1	0	1
0	0	1	0	0	0	0
0	1	0	1	1	1	1
0	1	1	1	0	1	0
1	0	0	1	1	1	1
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	1	1	1	1	1

Present State A	Present State B	Input x	Next State A	Next State B	D_A	D_B
0	0	0	0	1	0	1
0	0	1	0	0	0	0
0	1	0	1	1	1	1
0	1	1	1	0	1	0
1	0	0	1	1	1	1
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	1	1	1	1	1

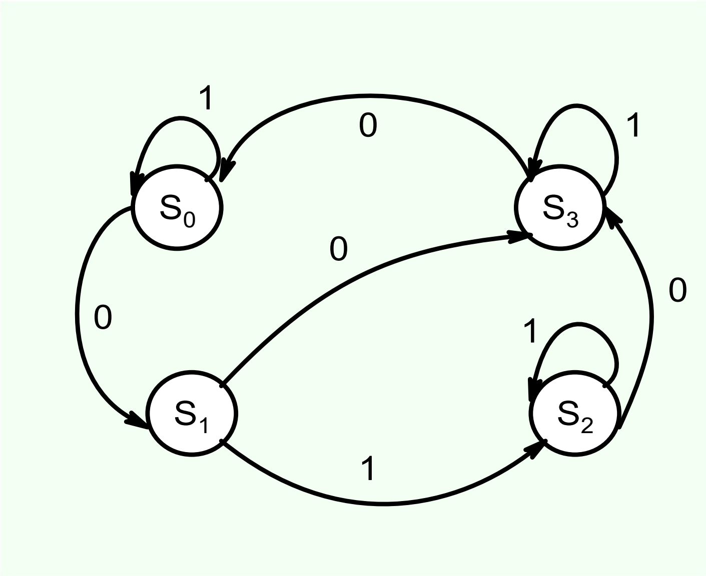


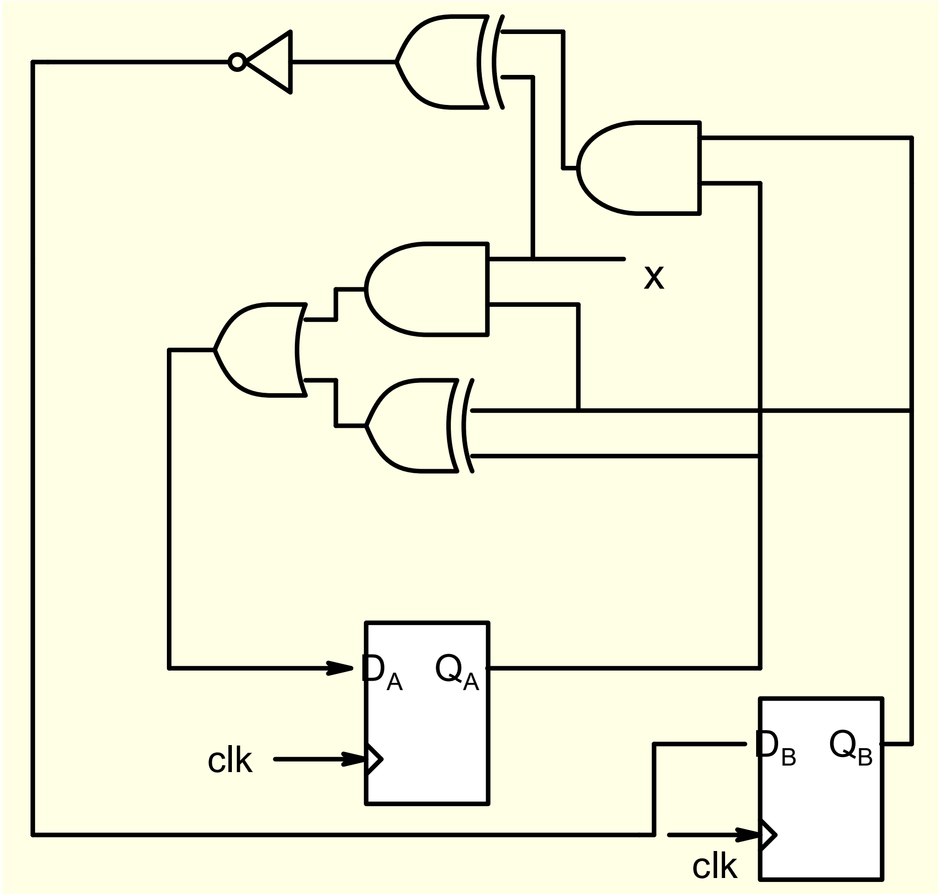
$$D_A = \overline{A}B + xB + A\overline{B}$$
$$= A \oplus B + x.B$$



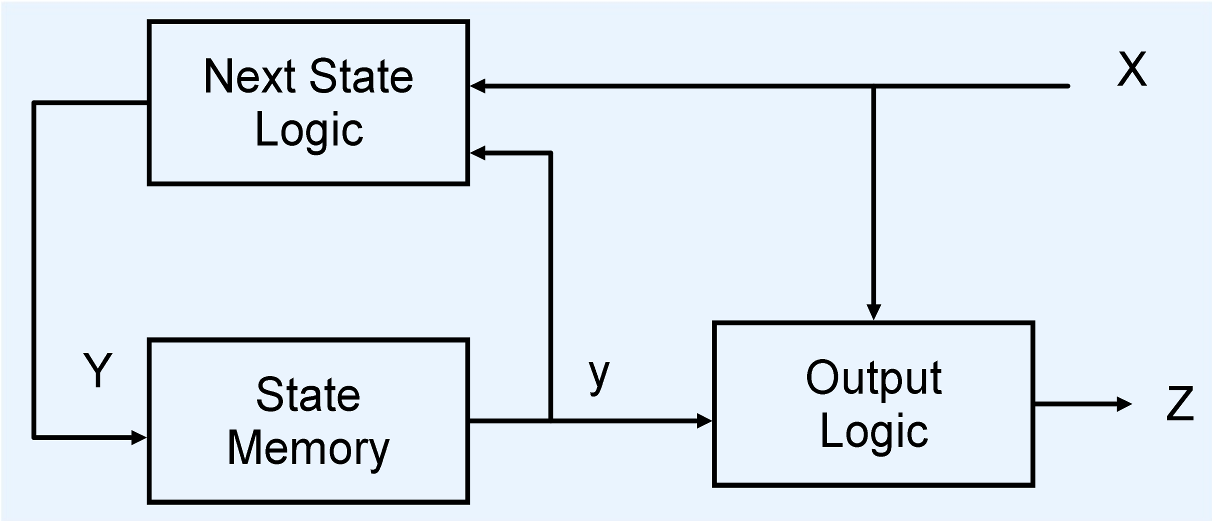
$$D_B = \overline{x}.\overline{A} + \overline{x}.\overline{B} + x.A.B$$
$$= \overline{x}.(\overline{A} + \overline{B}) + x.A.B$$
$$= \overline{x}.\overline{AB} + x.AB = \overline{x \oplus AB}$$

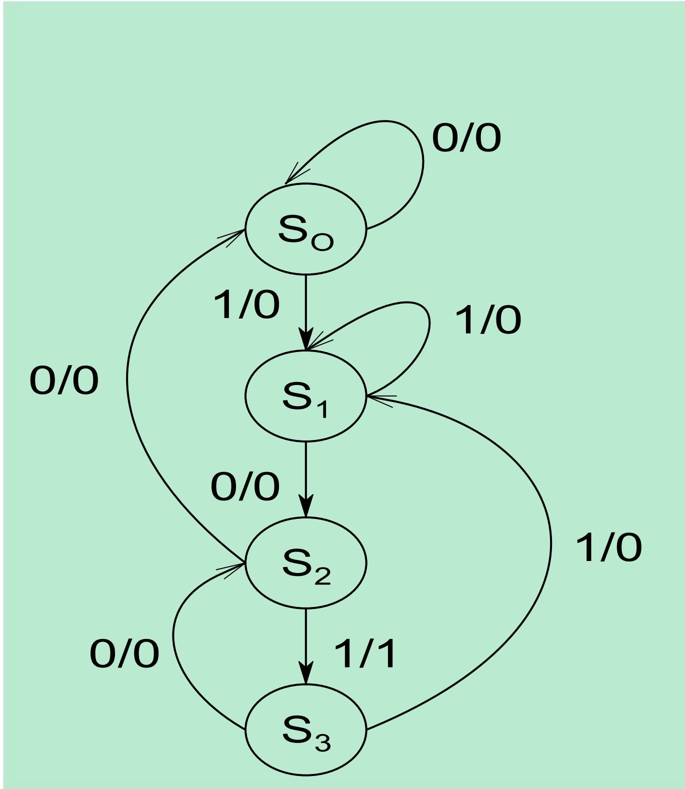
$$D_A = A \oplus B + x.B \qquad D_B = \overline{x \oplus AB}$$





Sequential Circuits: Summary





•we need to know how the system goes from one state to another in response to the inputs and how the outputs respond to these changes

Synthesis involves the following tasks:

- Number of storage elements
- State Encoding
- Choosing a flipflop type to implement states
- Synthesize next state logic
- Synthesize output logic